



Dhoom 4

Link submit: <https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/practice-problems/algorithm/dhoom-4/description/>

Solution:

C++	http://ideone.com/lnKrzK
Java	http://ideone.com/otLNxY
Python	http://ideone.com/Z6oZHN

Tóm tắt đề:

Để mở được ổ khóa, bạn cần có một chiếc chìa khóa có giá trị đúng bằng giá trị được ghi trên ổ khóa. Hiện tại bạn đang có trong tay một chìa khóa duy nhất có giá trị là X . Cho bạn N con số, khi bạn thực hiện nhân giá trị trên chìa khóa hiện có của mình với bất cứ số nào trong N con số được cho thì bạn nhận lại được một chiếc chìa khóa có giá trị bằng tích của chúng mod cho 100.000. Đây sẽ là chìa khóa mới của bạn và bạn sẽ dùng chìa khóa này cho những lần nhân tiếp theo.

Cụ thể, nếu số trên chìa khóa của bạn là X và bạn chọn nhân nó với số Y thì chìa khóa lúc này của bạn có giá trị là $(X * Y) \% 100.000$.

Cho rằng mỗi lần thực hiện phép nhân chìa khóa như vậy, ta chỉ tốn một giây và số lần nhân là không giới hạn. Hãy xác định thời gian tối thiểu để đạt được giá trị khóa cần tìm.

Input:

Dòng đầu tiên chứa hai số nguyên là số in trên chìa khóa của bạn và giá trị cần tìm.

Dòng tiếp theo chứa số nguyên N ($1 \leq N \leq 1000$) - số lượng con số bạn có thể chọn để thực hiện phép nhân chìa khóa.

Dòng cuối cùng gồm N số nguyên được cho. Giá trị của số được cho nằm trong khoảng $[1, 100.000]$.

Output:

Thời gian tối thiểu bạn cần để bạn đạt được giá trị khóa cần tìm. Nếu không thể đạt được in -1.

Ví dụ:

3 30 3 2 5 7	2
--------------------	---

Giải thích ví dụ:

Giá trị chìa khóa ban đầu của bạn là 3 và giá trị khóa là 30. Bạn có thể chọn nhân chìa khóa với một trong 3 số 2, 5 hoặc 7. Cách tối ưu để đạt được giá trị này như sau:

Lần 1: $3 \times 2 = 6$

Lần 2: $6 \times 5 = 30$

Hướng dẫn giải:

Nếu xem mỗi giá trị mà chìa khóa có thể nhận được là một đỉnh của đồ thị và mỗi phép nhân chìa khóa với một số bất kỳ trong N số được cho tạo nên một cạnh của đồ thị thì ta có thể sử dụng giải thuật BFS cho bài toán này như sau:

- Bước 1: Xác định đỉnh xuất phát là giá trị ban đầu của chìa khóa.
- Bước 2: Xác định đỉnh đích là giá trị khóa cần đạt được.
- Bước 3: Duyệt BFS từ đỉnh xuất phát. Mỗi lần lấy ra một đỉnh u từ hàng đợi, ta thực hiện phép nhân giá trị của đỉnh đó với lần lượt từng số trong N số được cho. Kết quả thu được từ phép nhân trên được mod cho 100.000 chính là một đỉnh v mới trong đồ thị. Bước làm này chính là ta đang phát sinh các cạnh (u, v) có thể có của đồ thị.

Lưu ý trong quá trình duyệt, ta sử dụng thêm một mảng dist[] với dist[v] là độ dài đường đi ngắn nhất từ đỉnh xuất phát đến đỉnh v. Như vậy khi ta tìm được một đỉnh v có giá trị đúng bằng giá trị đỉnh đích thì ta xuất ra ngay giá trị của dist[v].

Ngoài ra khi thực hiện phép nhân chìa khóa, ta phải cẩn thận vì có thể xảy ra trường hợp bị tràn số.

Độ phức tạp:

Time Complexity: Vì các đỉnh trên đồ thị, khi đã thăm rồi thì không được thăm lại nữa, do đó ta lấy tối đa chỉ có 100.000 đỉnh ra khỏi hàng đợi. Mặt khác, với từng đỉnh ta lấy ra khỏi hàng đợi, ta cần duyệt thêm N khóa nữa để có thể sinh ra các khóa mới. Như vậy, độ phức tạp thuật toán theo lý thuyết là $O(100.000 * N)$.

Space Complexity: $O(100.000)$.